Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Admir Ališić

Prenova in nadgradnja informacijskega sistema

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: doc. dr. Rok Rupnik

Ljubljana 2014

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Za informacijski sistem Islamske skupnosti v Republiki Sloveniji izdelajte analizo potrebnih sprememb, in sicer tako na tehnološkem področju kot tudi na področju funkcionalnosti. V okviru analize tehnološkega področja posvetite pozornost tudi analizi tehnologij, ki naj rezultira v seznam tehnologij, ki jih je treba nadomestiti s sodobnejšimi tehnologijami. Na podlagi analize izvedite prenovo informacijskega sistema.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Admir Ališić, z vpisno številko **63040189**, sem avtor diplomskega dela z naslovom:

Prenova in nadgradnja informacijskega sistema

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) in ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu prek univerzitetnega spletnega arhiva.

V Ljubljani, dne 20. septembra 2014 Podpis avtorja:

Zahvaljujem se profesorju doc. dr. Roku Rupniku za nasvete in pomoč pri celotnem procesu izdelave diplomskega dela. Posebno bi se rad zahvalil mami, ki me je spodbujala med študijem, in ženi, ki me je motivirala pri samem zaključku študija in izdelavi diplomskega dela.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
SQL	Structured Query Language	strukturirani povpraševalni jezik
HTTP	Hypertext Transfer Protocol	komunikacijski protokol
HTTPS	HTTP Secure	varen komunikacijski protokol
IIS	Internet Information Services	razširljiv spletni strežnik
JSON	JavaScript Object Notation	preprost format za izmenjavo podatkov
XML	Extensible Markup Language	razširljiv označevalni jezik
.NET	Microsoft Software Framework	Microsoftovo ogrodje za razvoj programske opreme
SSL	Secure Sockets Layer	kriptografski protokol za varno komunikacijo
TLS	Transport Layer Security	kriptografski protokol za varno komunikacijo
API	Application Programming Interface	programski vmesnik
CSR	Certificate Signing Request	zahtevek za izdajo certifikata

Povzetek

Diplomsko delo opisuje postopek prenove in nadgradnje obstoječega informacijskega sistema Islamske skupnosti v Republiki Sloveniji. Zajema analizo, načrtovanje in razvoj potrebnih sprememb v sistemu. Pri prenovi sistema sta se večinoma uporabljala orodje Microsoft Visual Studio 2010 in programski jezik `C#`. Predstavljena so tudi ostala orodja, tehnologije in metode, ki so del procesa prenove in nadgradnje sistema.

Za prenovo informacijskega sistema so se v islamski skupnosti odločili predvsem zaradi tehničnih problemov v obstoječem sistemu. Poleg tega so hoteli zvišati raven varnosti podatkov, dodati nove funkcionalnosti in vpeljati mobilno aplikacijo.

V diplomskem delu podrobneje predstavim prvo fazo prenove informacijskega sistema, ki zajema predvsem prenovo zaledja sistema in manjši del nadgradnje sistema, ki je viden uporabniku.

Ključne besede: prenova informacijskega sistema, zaledje sistema, podatkovna baza.

Abstract

The thesis deals with the process of renovating and upgrading the existing information system of the Islamic Community in the Republic of Slovenia. It involves analysing, planning and developing the necessary changes in the system. To this end, the majority of work was performed using Microsoft Visual Studio 2010 and the `C#` programming language. Moreover, the thesis presents other tools, technologies and methods that contributed to the renovation and upgrading of the system.

The Islamic Community opted for the renovation because they encountered difficulties in the existing system. They were also interested in raising the level of data security, add new functionalities and introduce a mobile application.

The thesis provides a detailed insight into the first renovation phase, which includes the renovation of the system back-end and a section of system upgrade that is visible to the user.

Keywords: information system renovation, system back-end, database.

Poglavje 1

Uvod

Leta 2010 smo za Islamsko skupnost v Republiki Sloveniji razvili informacijski sistem, kjer smo sodelovali pri razvoju zaledja sistema. Kot pri vsakem informacijskem sistemu se slabosti in ideje za popravke, nadgradnje in razširitve pokažejo ob uporabi samega sistema v realnem okolju. Tako se je po večletni uporabi sistema pojavila potreba po prenovi in nadgradnji obstoječe rešitve. Ideja je bila, da se odpravi tehnične napake, zaradi katerih je bilo treba občasno ročno poseganje in odpravljanje napak v sami informacijski rešitvi, poleg tega pa podpreti dodatne procese znotraj skupnosti. Delovanje skupnosti je bilo poenoteno z uvedbo trenutnega informacijskega sistema. Tako imajo vsi odbori, ki so razdeljeni po večjih slovenskih mestih, enoten vir podatkov, kar olajša tudi nadzor delovanja odborov samemu vrhu Islamske skupnosti v Republiki Sloveniji.

Prenovo in nadgradnjo sistema smo razdelili na tri faze, in sicer na prenovo in razširitev zaledja informacijskega sistema, na prenovo in poenostavitev obličja sistema ter na razvoj mobilne aplikacije. V diplomskem delu bom podrobneje predstavil prenovo in razširitev zaledja informacijskega sistema ter predstavil orodja, tehnologije in metode, ki sem jih uporabljal pri razvoju.

Poglavje 2

Okolje informacijskega sistema

Uporabnik informacijskega sistema je Islamska skupnost v Republiki Sloveniji.

2.1 Islamska skupnost v Republiki Sloveniji

Islamska skupnost v Republiki Sloveniji je javna, enkratna in samostojna verska skupnost vseh prebivalcev Republike Slovenije, ki sprejemajo islam za svojo vero. Zavzema se za duhovnost in človekovo dostojanstvo v zasebnem in javnem življenju, prizadeva si za osmišljanje bivanja na področju verskega življenja in ima s svojim delovanjem hkrati tudi pomembno vlogo v javnem življenju. Z razvijanjem svojih kulturnih, vzgojnih, izobraževalnih, solidarnostnih, dobrodelnih in drugih dejavnosti s področja socialne države bogati nacionalno identiteto in s tem opravlja pomembno družbeno vlogo [1].

Islamska skupnost v Republiki Sloveniji je splošno koristna, verska organizacija, ki svobodno in avtonomno uči svojo vero, opravlja verske obrede in posle, samostojno in svobodno oblikuje svojo organizacijsko strukturo, skrbi za versko in kulturno-izobraževalno dejavnost, rešuje finančna, administrativna, lastniška in druga vprašanja, povezana z delovanjem islamske skupnosti. Avtonomija in svoboda delovanja islamske skupnosti sta zagotovljeni s pravnim redom Republike Slovenije [1].

2.2 Organizacijska struktura skupnosti

Islamsko skupnost sestavljajo odbori kot osnovne organizacijske enote. Odbor je v tradicionalni organizacijski formi islamske skupnosti medžlis, a v organizaciji Islamske skupnosti v Republiki Sloveniji ustreza džematu. Odbor obsega najmanj 250 muslimanskih gospodinjstev na določenem področju, ki so med seboj povezana v izvrševanju skupnih islamskih dolžnosti. Odbor je pravna oseba zasebnega prava, vendar ne more delovati brez soglasja Mešihata islamske skupnosti. Pravno osebnost pridobi z registracijo pri Uradu Vlade Republike Slovenije za verske skupnosti z zahtevo, ki jo vloži Mešihat [1].

2.3 Članstvo v islamski skupnosti

Vsak odbor islamske skupnosti ima svoje člane in zaposlene. Član je lahko vsaka polnoletna oseba, ki sprejema islam za svoje versko prepričanje. Aktivni član skupnosti je vsak član, ki redno plačuje članarino. S tem postanejo člani islamske skupnosti tudi vsi člani njegove družine, skupaj pa se v evidenci vodijo kot gospodinjstvo.

Poglavje 3

Opredelitev problema

3.1 Predstavitev problema

Pri sami vpeljavi novega informacijskega sistema je bil sistem glede na trenutno stanje v islamski skupnosti zastavljen preobsežno s preveč funkcionalnostmi naenkrat, saj se je do tedaj posluževala ločenih evidenc znotraj vsakega odbora, ki ni bil poenoten. To je za povprečnega zaposlenega v islamski skupnosti pomenilo veliko obremenitev. Sistem je predvidel zajem velike količine podatkov, ki niso bili nujno potrebni za osnovno delovanje skupnosti in so posledično vnašali nepotrebne dvome uporabnika ob samem vnosu podatkov v sistem. Zaradi pomanjkanja ustreznega kadra znotraj skupnosti so se tako v nekaterih odborih angažirali bolj, v drugih manj.

Poleg vsebinskih in kadrovskih problemov se v sistemu pojavljajo tudi tehnični problemi. Odkrili smo kar nekaj napak pri vnosih podatkov v podatkovno bazo, vendar so bile te napake odkrite sproti, in smo jih čez celoten življenjski cikel aplikacije od uvedbe odpravljali sproti, in sicer ročno. S tem smo zagotovili, da so podatki konsistentni. Poleg problemov z vnosi so se pojavljali še problemi pri dodeljevanju pravic uporabnikom. Ti so bili posledica nerazumevanja med vodstvom skupnosti in razvojno ekipo pri zajemu zahtev pred razvojem informacijskega sistema. Zaradi pomanjkanja časa smo v prvi verziji sistema znižali prioriteto funkcionalnostim, kot sta delovanje aplikacije v varnem načinu SSL in beleženje napak

pri samem delovanju sistema. Posledica preložitve teh funkcionalnosti v naslednjo verzijo informacijskega sistema je bila ta, da je bil implementiran sedaj, štiri leta od uvedbe sistema.

Poleg odprave teh pomanjkljivosti se je pojavila tudi želja, da bi se v bodoče lahko uvozilo člane iz starih evidenc skupnosti, ki so shranjene v različnih elektronskih oblikah, in beleženje urnika verouka otrok. Tako bi se znebili razpršenih arhivov in poenostavili brskanje po njih. Zadnja želja je bila nadgradnja sistema, ki bi omogočala implementacijo mobilne aplikacije, s čimer bi tudi v islamski skupnosti sledili zadnjim trendom.

3.2 Cilji prenove in nadgradnje informacijskega sistema

Glavni cilj prenove sistema je odprava tehničnih napak in s tem posledično odprava ročnega poseganja v podatkovno bazo ter v sam sistem. Omogočiti smo želeli medsebojno pomoč zaposlenim med različnimi odbori, zato tudi potreba po spremembi načina dodeljevanja pravic. Ob vsem tem je pomembno, da se dvigne raven varnosti samih podatkov, saj gre za osebne podatke članov islamske skupnosti, za katere obstajajo zakonska določila o njihovi rabi in varovanju pred dostopom tretjih oseb. Cilj je torej še bolj poenostaviti in olajšati delo zaposlenim, kar je končni cilj vsakega informacijskega sistema.

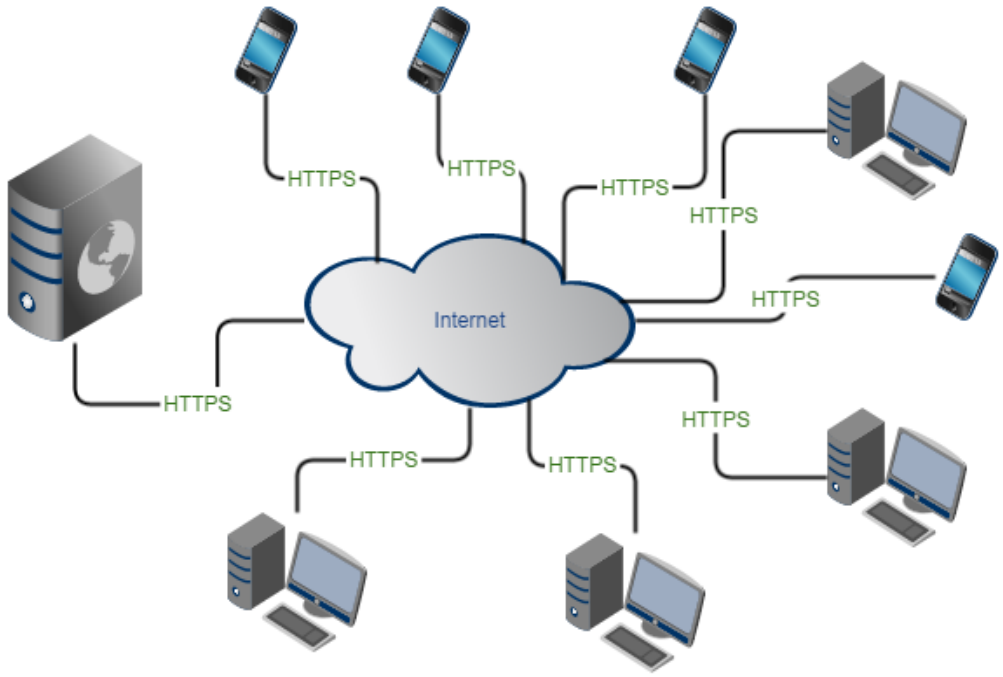
3.3 Predlagana rešitev

Z analizo smo prišli do ugotovitve, da je treba poseči na različne nivoje spletne aplikacije, in sicer tako v zaledje kot obličje. Ločeni del pa je implementacija mobilne aplikacije. Rešitev problema smo razdelili v tri faze po naslednjem vrstnem redu:

- prenova in razširitev zaledja sistema,
- prenova in poenostavitev obličja sistema,

- razvoj mobilne aplikacije.

Vse faze so precej lepo ločene in omogočajo zaporedni razvoj. Tako ni potrebe po sočasni implementaciji vseh faz. Odločili smo se, da začnemo pri zaledju sistema. Po tej fazi je mogoče izpeljati nadomestitev tega dela sistema v produkcijskem okolju. Zato smo se med drugim odločili, da iz obstoječe izvorne kode ločimo novo vejo in nadaljujemo razvoj sistema ločeno. To nam omogoča, da po zaključku faze primerjamo rezultate poizvedb na zaslonskih maskah ob predpostavki, da imamo na testni instanci aplikacije iste podatke kot na produkcijskem okolju. Predvideno je, da bo to tudi eden od pristopov testiranja pred samo uvedbo prenovljenega dela sistema v produkcijsko okolje. Po prenovi tega dela sistema je predvideno, da bo sistem deloval, kot je prikazano na Sliki 3.1.



Slika 3.1: Struktura informacijskega sistema po prenovi zaledja sistema.

Poglavje 4

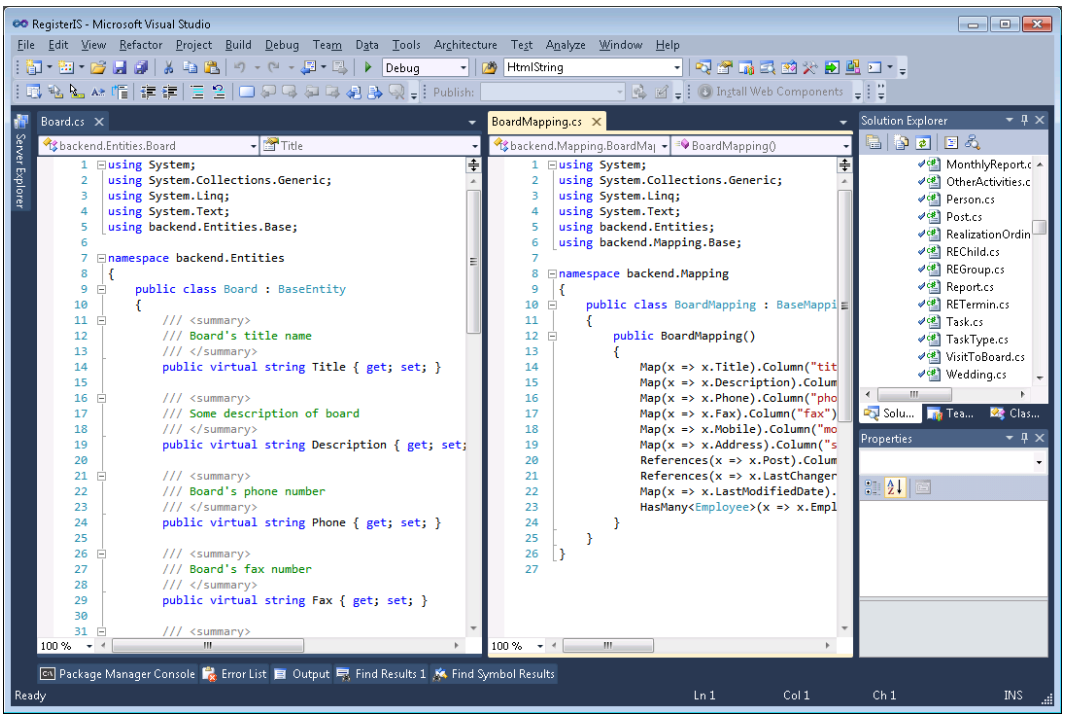
Uporabljena orodja in tehnologije

Pri prenovi in nadgradnji zaledja sistema so bila uporabljena naslednja orodja in tehnologije:

- Microsoft Visual Studio 2010 - razvojno okolje z uporabo programskega jezika `C#`,
- Postgre SQL - podatkovna baza,
- NHibernate - programska knjižnica za delo s podatkovno bazo,
- Fluent NHibernate - programska knjižnica za delo s podatkovno bazo,
- log4net - programska knjižnica za beleženje napak v sistemu,
- HTTPS - varna različica standarda http,
- IIS (Internet Information Services) - razširljiv spletni strežnik podjetja Microsoft,
- JSON (JavaScript Object Notation) - preprost format za izmenjavo podatkov,
- TortoiseSVN - programska oprema za nadzor verzij izvorne kode.

4.1 Visual Studio 2010

Microsoft Visual Studio 2010 je integrirano okolje, ki poenostavi celoten življenjski cikel razvijanja informacijske rešitve, in sicer od načrtovanja do uvajanja. V omenjenem okolju se lahko uporablja orodja za izdelovanje prototipov, modeliranje in vizualno načrtovanje, s katerimi razvijalec oživi svoje vizije. Nudi integrirano okolje, v katerem lahko razvijalci s svojim znanjem modelirajo, kodirajo, odpravljajo napake, preizkušajo in uvajajo vse več vrst programov. Visual Studio 2010 poenostavi pogosta opravila in razvijalcem omogoča raziskovanje v globino platforme. Ponuja zmogljiva orodja za upravljanje projekta, ohranjanje izvorne kode in iskanje napak. Preizkuševalci in razvijalci lahko uporabljajo ročno in samodejno preizkušanje ter napredna orodja za odpravljanje napak, in tako zagotovijo, da sestavljajo pravi program na pravi način [2].



Slika 4.1: Razvojno okolje Microsoft Visual Studio 2010.

4.2 Postgre SQL

Postgre SQL, pogosto tudi Postgres, je objektno relacijski sistem za upravljanje podatkovnih baz s poudarkom na razširljivosti in skladnosti s standardi. Kot podatkovni strežnik ima za glavno nalogo shranjevanje podatkov in kasnejše posredovanje le-teh na podlagi zahtevkov drugih programov oziroma aplikacij na istem računalniku ali prek mrežnih povezav (vključujoč internet) na podlagi zahtevkov programov oziroma aplikacij na drugem računalniku. Zmore obremenitve v razponu malih enoračunalniških aplikacij do velikih internetnih aplikacij z velikim številom hkratnih dostopov večjega števila uporabnikov. Zadnje različice ponujajo tudi podvajanje same podatkovne baze, kar pomeni večjo varnost in razširljivost [3].

4.3 NHibernate

NHibernate je odprtokodni objektno relacijski vmesnik za preslikavo, prirejen za ogrodje Microsoft .NET. Enostavno preslika objektne modele, izdelane s pomočjo Visual Studia v programskih jezikih `C#` ali VB.NET, kjer ni potrebe po posebnih implementacijah razredov ali atributov. V celoti podpira dedovanje in omogoča hiter razvojni cikel, saj lahko s preslikavo hitro ustvari tabele v podatkovni bazi iz domenskih objektnih modelov. Nudi podporo za vse bolj znane relacijske podatkovne baze in podpira razvoj najbolj zapletenih scenarijev [4]. NHibernate uporablja pri preslikavi objektov datoteke .hbm.xml, kar pomeni, da je treba vsako spremembo narediti na dveh mestih.

4.4 Fluent NHibernate

Fluent NHibernate ponuja alternativo NHibernateovim standardnim datotekam XML za preslikavo. Namesto pisanja datotek XML (.hbm.xml) vam Fluent NHibernate omogoča, da preslikavo napišete v kodi `C#`. To omogoča enostavnejše prestrukturiranje in boljšo berljivost kode. Ker datoteka XML ni v dosegu prevajalnika, se lahko zgodi, da ob preimenovanju lastnosti objekta znotraj razreda ta ni

posodobljena tudi v vaši datoteki za preslikavo, tega pa s samim prevajanjem kode ne boste zaznali. Napaka se bo pojavila šele v času izvajanja aplikacije. S tem, ko se preslikava dejansko premakne v kodo, je torej prevedena z ostalo programsko kodo, tako bo ob vsaki najmanjši spremembi kode prevajalnik ugotovil napako [5].

4.5 log4net

log4net knjižnica je Apachejevo orodje za pomoč razvijalcem pri beleženju dnevnika napak na različnih izhodih. log4net je različica odlične Apache log4j knjižnice, implementirane v ogrodju Microsoft .NET. Ohranjeno je ogrodje v duhu prvotne log4j, medtem ko so izkoriščene prednosti in nove funkcije ogrodja .NET [6].

4.6 HTTPS

HTTPS je varnejša različica standarda HTTP in se uporablja za elektronsko poslovanje in druge transakcije, kjer se prenašajo občutljivi osebni podatki. Uporablja SSL in TLS za kodiranje ter s tem zaščiti promet pred vmesnimi opazovalci, ki bi drugače lahko videli vsebino prenosa. Ta komunikacijski protokol navadno uporablja vrata številka 443. SSL (Secure Sockets Layer), ki je bil narejen za http, je posebno primeren, saj omogoča zavarovanje, tudi kadar prihaja informacija za kodiranje samo s strani strežnika. SSL uporablja kriptografski sistem, ki uporablja dva ključa za šifriranje podatkov. Prvi javni ključ je znan vsem, drugi pa je zasebni ključ in je znan le za prejemnika. Šifriranje SSL je edinstven in učinkovit način upravljanja podatkov v elektronskem poslovanju. Ko je digitalno potrdilo SSL nameščeno na spletni strani (strežniku), lahko uporabnik vidi ikono oziroma ključavnico na dnu območja navigatorja, naslov v naslovni vrstici pa se bo začel s HTTPS namesto HTTP, kar pomeni, da so podatki šifrirani [7].

4.7 Internet Information Services (IIS)

Internet Information Services (IIS, predhodno Internet Information Server) je razširljiv spletni strežnik, ustvarjen s strani Microsofta, za uporabo znotraj družine operacijskih sistemov Windows NT. Sestavni del je bil od Windows NT 4.0 naprej, v nekaterih različicah pa je bil odsoten (npr. izdaja Windows XP Home). Strežnik IIS ni privzeto vklopljen ob inštalaciji operacijskega sistema Windows [8].

4.8 JSON

JSON je preprost format za izmenjavo podatkov, ki je dokaj enostaven in lahko berljiv tako človeku kot računalniku. Zaradi teh lastnosti je JSON idealen za izmenjavo podatkov med strežnikom in odjemalcem [9].

4.9 TortoiseSVN

TortoiseSVN je brezplačni program za razvijalce (programerje). Razvijalcem pomaga nadzorovati različne verzije izvorne kode njihovih programskih rešitev. TortoiseSVN je klient celotne rešitve za nadzor kode in ni operativen brez strežniškega dela Subversion. Celotna rešitev je produkt programske fundacije Apache [10, 11].

Poglavje 5

Prenova in razširitev zaledja informacijskega sistema

5.1 Analiza obstoječe rešitve

Med analizo obstoječe rešitve smo se osredotočili na izkušnje uporabnikov pri dosedanjem delu. Tako smo prišli do naslednjih rezultatov analize in predvideli naslednje spremembe ter razširitve v informacijskem sistemu:

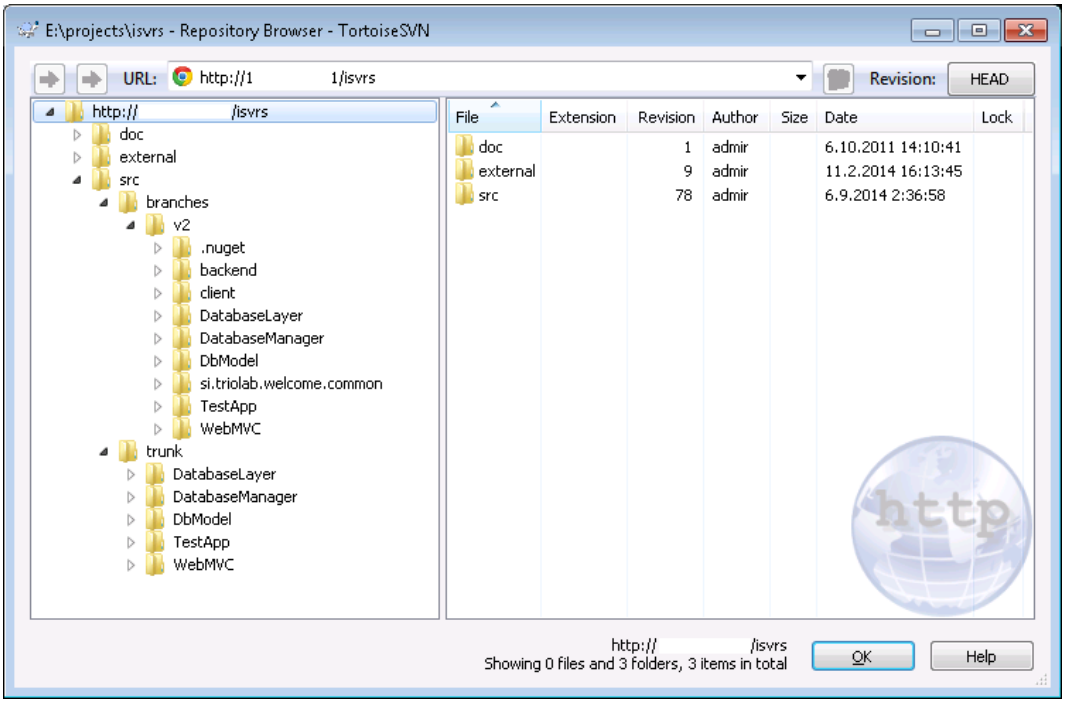
- ločitev dela implementacije, ki skrbi za preslikavo objektov v tabele podatkovne baze v ločeno knjižnico,
- ločitev implementacije vmesnika za upravljanje s podatkovno bazo v ločeno knjižnico,
- obvezna vpeljava transakcij pri izvajanju kompleksnejših ukazov nad podatkovno bazo,
- predelava in razširitev podatkovne baze na dveh delih:
  - dodeljevanje pravic uporabnikom (predelava),
  - urniki verouka in beleženje prisotnosti otrok (razširitev);

- vpeljava beleženja napak na različnih nivojih informacijske rešitve za lažji nadaljnji razvoj in vodenje dnevnika napak,
- implementacija varne povezave HTTPS,
- plan razširitve sistema za podporo bodočemu razvoju mobilne aplikacije,
- razvoj API (Application Programming Interface), ki bo podlaga za implementacijo mobilne aplikacije,
- izbira formata in priprava predloge datoteke za poenoten uvoz podatkov iz starih evidenc,
- implementacija razširitve za uvoz podatkov iz starih evidenc na podlagi prej izdelane predloge datoteke.

Po implementaciji pa je treba opraviti še teste in primerjati rezultate stare in nove rešitve na realni podatkovni bazi. Zaradi predelave podatkovne strukture smo morali biti previdni tudi pri uvozu podatkov v novo predelano bazo podatkov. Za testiranje je bila predvidena uporaba celotnega informacijskega sistema, se pravi tudi obstoječih zaslonskih mask. Ker pa naj bi primerjali podatke z obstoječim sistemom, je bila predvidena postavitev celotnega testnega okolja z uvozom produkcijske baze. Po uspešno opravljenih testiranjih pa je sledila še zamenjava obstoječega sistema z novimi posodobljenimi funkcionalnostmi v produkcijskem okolju.

5.2 Ustvarjenje nove veje kode

Pred začetkom kakršnihkoli sprememb smo ustvarili ločeno vejo izvorne kode in nadaljevali razvoj na tej veji kode. To nam je kasneje omogočalo vzporedno primerjavo rezultatov poizvedb stare in nove rešitve. Na Sliki 5.1 lahko vidite drevesno strukturo map, kjer je vidna ločena veja z dodatnimi projekti znotraj mape v2.



Slika 5.1: Programska oprema za nadzor izvorne kode TortoiseSVN.

5.3 Prestrukturiranje in razširitev baze podatkov

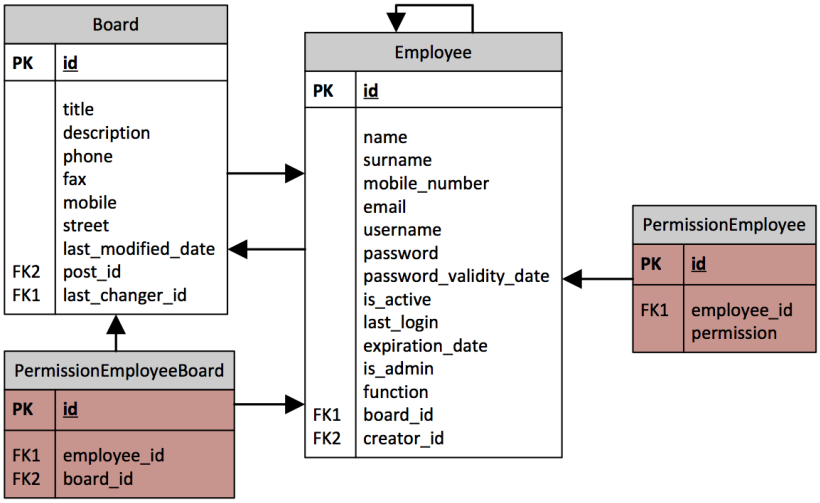
Ko govorimo o podatkovni bazi in njenem modeliranju, največkrat pomislimo na orodje, s katerim lahko ustvarjamo tabele, dodajamo stolpce tabel in definiramo tip podatka, ki ga predstavlja stolpec. V našem primeru je bil pogled na bazo podatkov in njeno načrtovanje s perspektive objektov. Tako se prestrukturiranje in razširitev podatkovne baze prepleta z implementacijo ločene knjižnice (glej podpoglavje 5.4). Fluent NHibernate nam omogoča, da s pomočjo razredov za preslikavo definiramo, kakšna naj bo videti tabela znotraj podatkovne baze. Prav tako pa nam, ko imamo vse preslikave definirane z razredi, omogoča avtomatsko generiranje podatkovne baze. Velika večina tabel v podatkovni bazi je ostala nespremenjena, tako smo le prepisali obstoječe objekte in njihove preslikave. Spremembe so se pojavile pri

naslednjih funkcionalnostih sistema:

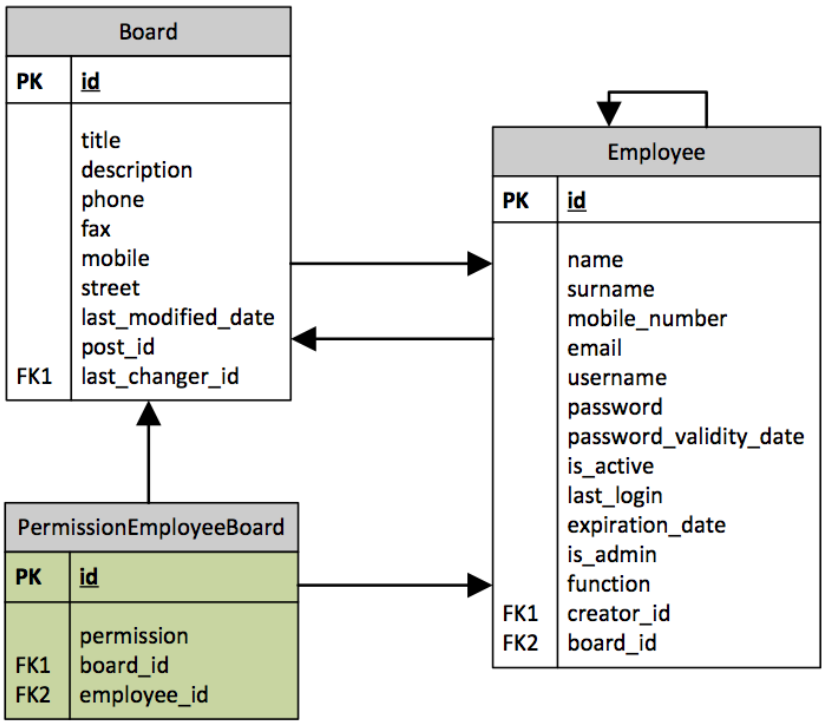
- dodeljevanje pravic uporabnikom (predelava),
- urniki verouka in beleženje prisotnosti otrok (razširitev).

5.3.1 Predelava podatkovnega modela za dodeljevanje pravic

Problem, ki smo ga morali rešiti pri dodeljevanju pravic zaposlenim, je bil, da podatkovna struktura ni dovoljevala, da bi zaposleni lahko opravljal samo določeno nalogo v enem odboru in določeno nalogo v drugem odboru. V prvi verziji podatkovnega modela za dodeljevanje pravic smo imeli dve ločeni tabeli. V eni smo hranili povezavo med pravico in zaposlenim, v drugi pa povezavo med zaposlenim in odborom. To je pomenilo, da smo ob povezavi zaposlenega na odbor določili le, da lahko opravlja funkcije, ki jih je imel, določene v tabeli za pravice, se pravi, da so mu bile za vse odbore dodeljene enake pravice. V prenovljenem modelu smo tabelo za pravice združili s tabelo za povezavo zaposlenega z odborom in dodali dodatni atribut, ki določa, za katero pravico gre. Na Slikah 5.2 in 5.3 si lahko ogledate podatkovni model pred predelavo podatkovnega modelain po njej. Podatkovni tabeli na Sliki 5.2, ki sta bili združeni v eno samo tabelo, sta označeni z rdečo barvo ozadja, podatkovna tabela, ki je nastala kot rezultat združitve, pa je na Sliki 5.3 označena z zeleno barvo ozadja.



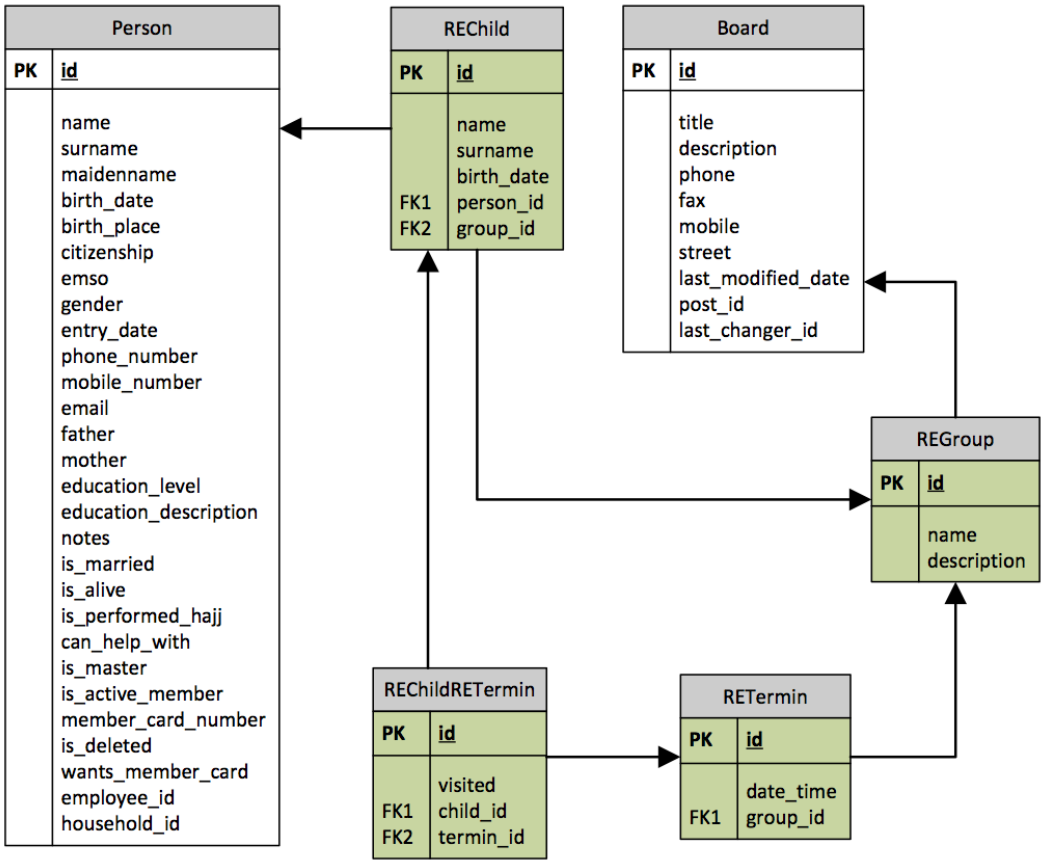
Slika 5.2: Podatkovna struktura tabel za dodeljevanje pravic pred prenovo.



Slika 5.3: Podatkovna struktura tabel za dodeljevanje pravic po prenovi.

5.3.2 Razširitev podatkovnega modela urnik verouka

Pri zajemu zahtev je bila želja predstavnikov skupnosti, da bi jim v sistemu omogočili vodenje urnikov in prisotnosti otrok pri verouku. Na podlagi analize smo kot rezultat razširitve pripravili podatkovno strukturo, ki je prikazana na Sliki 5.4.



Slika 5.4: Razširitev podatkovnega modela za potrebe verouka.

Na Sliki 5.4 lahko vidimo z zeleno označene novo nastale podatkovne tabele. Te opisujejo naslednja dejstva, ki so bila posledica analize:

- otrok, ki obiskuje verouk, je lahko član nekega gospodinjstva in v tem primeru je vezan na osebo, ki je že v sistemu,
- v nasprotnem primeru se lahko za otroka, ki obiskuje verouk, vnese le priimek, ime, rojstni datum in skupino v katero je vpisan,
- skupini lahko določimo ime in opis. Predvideno je, da z imenom in opisom skupine določijo stopnjo znanja, ki ga pridobijo v tej skupini,
- vsaka skupina ima lahko poljubno število terminov,
- vsak otrok ima lahko povezavo na termine, ki bi jih moral obiskati. Poleg tega je možno pri vsaki povezavi otroka in termina določiti, ali je bil na tem terminu prisoten ali ne.

5.4 Implementacija ločene knjižnice

Implementacija ločene knjižnice se je krepko prepletala s prestrukturiranjem in razširitvijo podatkovne baze zaradi pristopa k bazi s preslikavo objektov v tabele. Tako smo poleg prestrukturiranja in razširitve podatkovne baze izvedli naslednje:

- preslikave objektov v tabele podatkovne baze,
- implementirali vmesnik za upravljanje s podatkovno bazo,
- vpeljali transakcije na vmesniku za upravljanje s podatkovno bazo.

Pred samim začetkom implementacije nove knjižnice je bilo treba pregledati obstoječo rešitev. Našli smo kar nekaj stvari, ki bi jih lahko izboljšali, in se na podlagi teh odločili, da namesto obstoječih več razredov, ki so imeli implementirane ukaze za izvajanje, nad podatkovno bazo združimo vse metode v dva razreda, in sicer enega, ki vsebuje metode za branje, in drugega, ki vsebuje metode za pisanje. Pri implementaciji metod za pisanje smo uvedli transakcije. Za njihovo uvedbo

smo se odločili zaradi napak, ki so se dogajale, ko je bilo treba v enem klicu izvesti dve ali več operacij v različne tabele znotraj podatkovne baze.

5.4.1 Transakcije

Transakcije so mehanizem za skupinske operacije v tem smislu, da uspejo vse ali pa nobena. To prepreči probleme v podatkovni bazi, ki bi nastali, če bi se uspešno izvedel le del operacij [12].

Primer v obstoječem informacijskem sistemu:

Ob dodajanju novega gospodinjstva v podatkovno bazo je bilo treba vnesti tudi nosilca gospodinjstva. V primeru, da je šlo obenem tudi za novo osebo, ki je morala biti na novo dodana v sistem, se je dogajalo (zaradi ne dovolj dobre preverbe vnosnih polj na vnosnih maskah), da se je oseba že shranila v podatkovno bazo, pri shranjevanju objekta gospodinjstvo pa je bilo zaradi manjkajočega obveznega podatka shranjevanje neuspešno. Posledica je bila ta, da je bila kljub obvestilu, da shranjevanje v podatkovno bazo ni uspelo, oseba shranjena. Ob novem uspešnem poskusu vnosa je bila nato oseba ponovno shranjena, in tako smo dobili podvojen vnos.

5.5 Beleženje napak v sistemu

Za lažje in bolj sistematično sledenje napakam pri delovanju informacijskega sistema smo uporabili knjižnico log4net. Sama knjižnica omogoča konfiguracijo in implementacijo na različne možne načine. Razvili smo razred, ki omogoča beleženje napak v tekstovno datoteko. Datoteka nam služi kot dnevnik napak.

Primer uporabe v kodi `C#`*:*

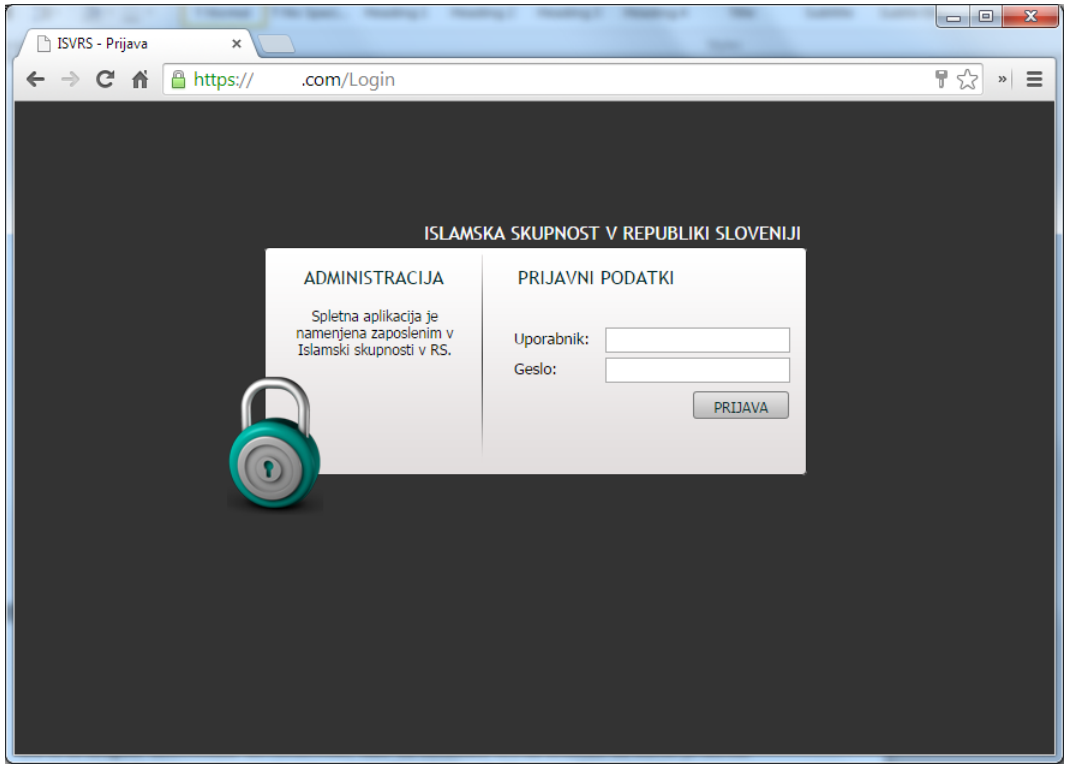
```
catch (Exception ex)
{
    Logger.Instance.WriteErrorMsg(ex);
    return new ApiJsonResultError(ex.Message);
}
```

5.6 Varna povezava HTTPS

Pri vpeljavi varne povezave je bilo treba najprej pridobiti certifikat SSL. Odločili smo se uporabiti brezplačni certifikat StartSSL izdajatelja StartCom. Pri implementaciji je bil potreben minimalen poseg v programsko kodo, in sicer smo na vsakem kontrolnem razredu dodali atribut RequireHttps (glej primer spodaj). To pomeni, da sistem ne dopušča izvedbe klica, če ta ni prišel prek url naslova, ki se začne s https://*.

```
#if !DEBUG
[RequireHttps]
#endif
```

Da bi se izognili morebitnim nevšečnostim, smo na strežniku IIS za domeno, na kateri je dostopen sistem, dodali preusmeritev vseh zahtev http://* na https://*. Na strežniku je bilo treba ustvariti zahtevek za izdajo certifikata (CSR). Ko je bil zahtevek ustvarjen, smo ga posredovali izdajatelju, ta pa nam je na podlagi zahtevka izdal certifikat, ki ga je bilo treba le še uvoziti med certifikate strežnika IIS. V zadnjem koraku smo le še povezali certifikat SSL z domeno. Na Sliki 5.5 lahko vidite, kakšna je videti naslovna vrstica v brskalniku pri dostopu do informacijskega sistema.



Slika 5.5: Informacijski sistem, dostopen prek varne povezave HTTPS.

5.7 Razvoj API za potrebe mobilne aplikacije

Ker pri samem zajemu zahtev vodilni v skupnosti niso bili prepričani, kaj točno želijo od mobilne aplikacije, je bil dogovor, da za začetek postavimo ogrodje, ki bo vračalo podatke iz sistema v formatu JSON. Ker je bil dogovor, da se mobilna aplikacija implementira v tretji fazi, ni bilo pričakovanj, da bomo že v tej fazi implementirali popolnoma vse funkcionalnosti, ki jih podpira spletna aplikacija. Tako smo v prvi verziji API-ja razvili naslednje:

- vračanje seznama odborov,
- vračanje podatkov o odboru,
- vračanje seznama zaposlenih v določenem odboru,

- vračanje podatkov o zaposlenem,
- vračanje seznama pravic zaposlenega,
- vračanje seznama gospodinjstev,
- vračanje podatkov o gospodinjstvu,
- vračanje seznama gospodinjstev glede na vnos iskalnega niza,
- vračanje seznama vseh oseb,
- vračanje seznama oseb glede na vnos iskalnega niza,
- vračanje seznama oseb določenega gospodinjstva,
- vračanje podatkov o osebi,
- vračanje podatkov o osebi glede na vnos številke članske izkaznice,
- vračanje seznama vplačanih članarin,
- vračanje podatkov o vplačani članarini,
- vračanje seznama donacij,
- vračanje podatkov o donaciji.

Primer odgovora strežnika v formatu JSON:

```
{
    success: "ok",
    result: {
      Id: 1,
      Title: "ODBOR LJUBLJANA",
      Phone: "+386(1)2313625",
      Fax: "+386(1)2313626",
    }
}
```

5.8 Uvoz obstoječih podatkov

Ob analizi za uvoz obstoječih podatkov smo prišli do zaključka, da je najpomembnejši uvoz starih članov, ki so se vodili le v ročni evidenci. Vsi odbori so v evidenci vodili le nosilca gospodinjstva brez družinskih članov in le tri osnovne podatke, in sicer ime, priimek ter naslov nosilca. Tako ni bilo težko izbrati format, ki bo uporabljen za uvoz. Izbrali smo tekstovno obliko CSV, kot ločilo pa je bilo izbrano podpičje.

Primer datoteke in formata za uvoz:

```
Admir;Alisic;Triglavska ulica;1;1235;Radomlje
Amar;Alisic;Triglavska ulica;1;1235;Radomlje
```

Da bi preprečili podvajanje zapisov, smo pri implementaciji metode uvoza definirali ključ, ki pomeni podvojitev vnosa, in sicer so ga sestavljali vsi podatki (ime, priimek in celoten naslov). V primeru, da se je oseba preselila na drug naslov in je slučajno že bila v sistemu, pa je bilo dogovorjeno, da se bo v primeru istega imena in priimka opravil ročni pregled v sistemu in se posledično ob ugotovitvi podvojene osebe naredi ročni izbris takega vnosa.

5.9 Testiranje

Testiranje je eden pomembnejših delov prenove sistema, saj si nismo smeli privoščiti večjih napak, še posebej pri vnosu podatkov. Tako smo se pri samem načrtovanju testiranja odločili, da bomo ločeno testirali vnos podatkov in ločeno poizvedbe raznih seznamov ter podrobnosti o posameznih zapisih v podatkovni bazi. Predvideli smo, da bodo rezultati pri testiranju novih vnosov najbolj pregledni na prazni podatkovni bazi, rezultati poizvedb pa na polni produkcijski bazi, kjer lahko opravljamo primerjavo med obstoječo rešitvijo in prenovljeno rešitvijo. Pred samim testiranjem smo tako pripravili testno okolje z novo strukturo podatkovne baze in najprej opravili teste vnosov na prazni bazi, zaradi česar je bilo testiranje tega dela

bolj enostavno in pregledno. Pri prenovi smo uporabili podobne testne scenarije kot pri samem razvoju aplikacije. Primere testnih scenarijev si lahko ogledate v podpoglavjih 5.9.1 in 5.9.2.

5.9.1 Vnosi novih zapisov v sistem

Testiranja novih zapisov smo se lotili v testnem okolju s prazno podatkovno bazo. To nam je omogočalo enostavnejši pregled nad vnesenimi podatki, saj smo pri vsakem vnosu vedeli, kaj se mora pojaviti v sami bazi podatkov. Tako smo poleg pregleda podatkov na zaslonskih maskah podatke pregledovali tudi direktno v tabelah podatkovne baze.

Primer testnega scenarija za vnos nekaj različnih entitet v sistem:

Št.	**Opis**	**Pričakovan rezultat**	**Test uspel**	**Izvajalec (datum)**
1	Pri vnosu novega gospodinjstva vnesemo vse podatke.	Aplikacija shrani vse podatke gospodinjstva.	DA	Admir (5.8.2014)
2	Pri vnosu novega gospodinjstva vnesemo vse podatke razen nosilca gospodinjstva.	Aplikacija nas opozori, da ni bilo mogoče shraniti gospodinjstva, ker manjka obvezni podatek - nosilec gospodinjstva.	DA	Admir (5.8.2014)

3	Pri vnosu novega gospodinjstva vnesemo vse podatke razen vnosnega polja "Priimek", pod katerim se vodi gospodinjstvo.	Aplikacija nas opozori, da ni bilo mogoče shraniti gospodinjstva, ker manjka obvezni podatek "Priimek".	NE, sistem dovoli vnos gospodinjstva.	Admir (5.8.2014)
4	Pri vnosu članarine izpolnimo vse podatke.	Aplikacija shrani vse podatke o članarini.	DA	Admir (5.8.2014)
5	Pri vnosu članarine ne vnesemo zneska.	Aplikacija nas opozori, da manjka obvezen podatek "Znesek članarine".	DA	Admir (5.8.2014)

5.9.2 Primerjava rezultatov na produkcijskih podatkih

Po testiranju vnosov novih zapisov v sistem je bilo treba podatkovno bazo izprazniti in uvoziti obstoječe podatke iz produkcijske baze. Tako smo omogočili primerjavo rezultatov na realnih podatkih.

Primer scenarija testiranja rezultatov različnih seznamov zaslonskih mask:

Št.	**Opis**	**Pričakovan rezultat**	**Test uspel**	**Izvajalec (datum)**
1	Seznam gospodinjstev sortiramo po datumu včlanitve. Vrnjenih je 447 gospodinjstev.	Aplikacija na testnem okolju s prenovljenim zaledjem sistema vrne isto število in sortirana gospodinjstva po datumu vnosa.	NE, zaradi napake pri implementaciji sortiranja.	Admir (12.8.2014)
2	Seznam vseh članov. Seznam vrne 2681 članov.	Aplikacija na testnem okolju s prenovljenim zaledjem sistema vrne isto število oseb.	DA	Admir (12.8.2014)
3	Seznam članov sortiramo po številki članske izkaznice. Vrnjenih je 2681 članov.	Aplikacija na testnem okolju s prenovljenim zaledjem sistema vrne isto število in sortirane člane po številki članske izkaznice.	NE, zaradi napake pri implementaciji sortiranja.	Admir (12.8.2014)
4	Vnos iskalnega niza pri iskanju osebe nam sistem vrne osebe, ki ustrezajo temu iskalnemu nizu.	Aplikacija na testnem okolju nam mora ob istem iskalnem nizu vrniti isto število oseb ter v istem vrstnem redu.	DA	Admir (12.8.2014)

5.10 Uvedba prenovljene rešitve

Sama uvedba prenovljenega zaledja aplikacije v produkcijsko okolje ni bila težavna. Razlog za to je predvsem v tem, da na uporabnike spremembe v tej fazi nadgradnje oziroma prenove niso vplivale, saj jim je aplikacija na prvi pogled videti takšna kot pred spremembami. Izjema je le naslovna vrstica v brskalniku, ki ima sedaj lep zeleno obarvan začetek url naslova HTTPS in zeleno ključavnico pred njim. Tako smo pri zamenjavi starega sistema z novim začasno onemogočili dostop do produkcijskega okolja in izvedli migracijo podatkovne baze na novo podatkovno bazo. Zaradi potreb testiranja je bil ta del naloge dokaj enostaven, saj smo že točno vedeli, katere tabele je treba združiti in na kakšen način. Ko smo enkrat prenesli podatke, je bila potrebna le še zamenjava prevedenega paketa kode v produkcijsko mapo strežnika.

Poglavje 6

Sklepne ugotovitve

V diplomskem delu sem predstavil prenovo in nadgradnjo informacijskega sistema. Veseli me, da sem lahko delal na sistemu, ki je dejansko v uporabi in bo skupnosti služil še mnoga leta. Čeprav sem sodeloval že pri kar nekaj različnih projektih, sem se tudi tu naučil nekaj novih stvari. Predvsem bi tu izpostavil implementacijo in konfiguracijo varnega načina prenosa podatkov. Ker sem sodeloval že pri prvotnem razvoju sistema, mi poseganje v izvorno kodo sistema in sama predelava nista povzročala večjih problemov. Je pa bila pomembna ugotovitev, da ima velik pomen sama komunikacija z naročnikom. Zelo pomembno je bilo, da je bila pred razvojem narejena dobra analiza obstoječega sistema in posledično je nastal tudi dober plan za nadaljnji razvoj. Ti dve stvari sta bili ključni, da je razvoj nato potekal brez večjih težav. Tako so bile največje težave pri sami prenovi narediti dobro analizo in plan razvoja ter novosti, in sicer vpeljava knjižnice za beleženje napak ter implementacija varne povezave. Pri slednji je bilo največ raziskovanja, kako in kje priti do certifikata za preverjanje domene. Z izdelavo diplomskega dela sem tako pridobil nova praktična znanja, za katera sem prepričan, da mi bodo v bodoče zelo koristila.

Literatura

[1] Statut Islamske skupnosti v Republiki Sloveniji, Ljubljana, Islamska skupnost v Republiki Sloveniji, 2010, str: 5, 35.

[2] (2014) Microsoft Visual Studio 2010. Dostopno na: http://www.microsoft.com/business/smb/sl-si/strezniki-in-orodja/visual-studio-pro.mspx

[3] (2014) Postgre SQL - Wikipedia, prosta enciklopedija. Dostopno na: http://en.wikipedia.org/wiki/PostgreSQL

[4] (2014) NHibernate Forge. Dostopno na: http://nhforge.org/

[5] (2014) Fluent NHibernate in a Nutshell. Dostopno na: https://github.com/jagregory/fluent-nhibernate/wiki/Getting-started

[6] (2014) What is Apache log4net™. Dostopno na: http://logging.apache.org/log4net/

[7] (2014) Kaj je HTTPS? Dostopno na: http://www.avelpo.si/index.php?option=com_content&task=view&id=43

[8] (2014) Internet Information Services - Wikipedia, prosta enciklopedija. Dostopno na: http://en.wikipedia.org/wiki/Internet_Information_Services

[9] (2014) Uvod v JSON. Dostopno na:
http://json.org/json-sl.html

[10] (2014) TortoiseSVN - Wikipedia, prosta enciklopedija. Dostopno na:
http://en.wikipedia.org/wiki/TortoiseSVN

[11] (2014) Apache Subversion - Wikipedia, prosta enciklopedija. Dostopno na:
http://en.wikipedia.org/wiki/Apache_Subversion

[12] (2014) Transakcije. Dostopno na:
http://colos.fri.uni-lj.si/eri/racunalnistvo/PODATKOVNE_BAZE/
transakcije.html